

Supplementary Manual 1 - MATLAB basics

The following introductory material and online tutorials available at <https://mathworks.com/help/matlab/getting-started-with-matlab.html> are recommended to learn some of the most useful features of MATLAB. In order to execute MATLAB in a Unix environment, at the prompt, type

```
$ matlab
```

and return. Alternatively, double-click on the MATLAB icon in your installation folder or the desktop (or dock). In MATLAB, each function has the following general form

```
>> [output1, ~, output3] = functionName(input1, input2)
```

where `input1` and `input2` denote mandatory inputs supplied to a function named `functionName`. The tilde character `~` denotes a placeholder for an output that is not stored in the workspace. The significant inequality sign `>>` denotes the MATLAB command line; anything directly following `>>` and before a semi-colon is meant to be either entered on the MATLAB command line or included in a script that implements a sequence of functions in an analysis pipeline. Such a script may be standard `.m` file, but preferably a MATLAB live script https://mathworks.com/help/matlab/matlab_prog/what-is-a-live-script.html, which is an interactive document that combines MATLAB code with embedded output, formatted text, equations, and images in a single environment. In this manuscript, all MATLAB code and commands are typed in monospaced characters. The COBRA Toolbox, as well as MATLAB, are documented extensively. For instance, more information on `commandName` can be retrieved by typing:

```
>> help commandName
```

Basic Commands

Load a file *filename.mat*:

```
>> load(filename.mat)
```

Save the workspace in *filename* (as *.mat* file):

```
>> save filename
```

Save only the variable in *filename* (as *.mat* file):

```
>> save filename variable
```

Clear workspace:

```
>> clear
```

Delete *only one variable*:

```
>> clear variable
```

Call *.m* file which contains a MATLAB script:

```
>> scriptname
```

Matrix-related commands

Create matrix *x* with 3 columns and 2 rows. The elements of this matrix are denoted $x(i, j)$, where *i* are the rows and *j* are the columns:

```
>> X = [1 1 1; 1 1 1]
```

Create vector with words (*name1*, *names2*) as entry:

```
>> names = {'name1' 'name2'}
```

Calculate the rank of matrix *x*:

```
>> rank(X)
```

Calculate the null space of matrix *x*:

```
>> null(X)
```

Calculate the singular value decomposition of matrix *x* where *u* contains the left singular vectors, *s* contains singular values, and *v* contains the right singular vectors:

```
>> [U, S, V] = svd(X)
```

Basic Graphic commands

Open a graphic window:

```
>> figure
```

Plot vector *x* versus *y* (type `help plot` to get more option information):

```
>> plot(x, y)
```

Create histograms of matrix x (*optional*: `hist(X, bins)` where `bins` is a number of how many points are averaged (default 100)):

```
>> hist(X)
```

Label the x-axes of the plot with *name*:

```
>> xlabel('name')
```

Label the y-axes of the plot with *name*:

```
>> ylabel('name')
```

Add title *name* on plot:

```
>> title('name')
```

Plot multiple graphs in one figure:

```
>> hold on
```

Disable multiple graphs:

```
>> hold off
```